```
import pandas as pd

Data2 = pd.read_csv("Data2.csv")

Data2


def sim1(a,b):

  if a*b < 0:

    return 0

  return min(a,b)/max(a,b)


def dissim1(a,b):

  return 1-sim1(a,b)


l = input().split(' ')

n = len(l)

seq = []

for i in l:

  seq.append(float(i))
```

```
def simhelper1num(seq):
  resnum = 0
  for i in range(len(seq)):
    if i == 0:
      resnum += seq[i] * sim1(seq[i],seq[i])
    else:
      for j in range(0,i+1):
        if (seq[j]<=seq[i]) :
          resnum += (seq[i] * sim1(seq[i],seq[j]))
        else :
          resnum += (seq[i] * -1 * sim1(seq[i],seq[j]))
  return resnum


def simhelper1den(seq):
  resden = 0
  for i in range(len(seq)):
      if i==0:
        resden += sim1(seq[i],seq[i])
```

```
    else:
      for j in range(0,i+1):
        resden += sim1(seq[i],seq[j])
  return resden


def fynplus1sim(seq):
  return (simhelper1num(seq) / simhelper1den(seq))


def dissimhelper1num(seq):
  resnum = 0
  for i in range(len(seq)):
    if i == 0:
      resnum += seq[i] * dissim1(seq[i],seq[i])
    else:
      for j in range(0,i+1):
        if ( seq[j]<=seq[i]):
          resnum += (seq[i] * dissim1(seq[i],seq[j]))
        else:
```

```
        resnum += (seq[i] * -1 * dissim1(seq[i],seq[j]))
  return resnum


def dissimhelper1den(seq):
    resden = 0
    for i in range(len(seq)):
      if i==0:
        resden += dissim1(seq[i],seq[i])
      else:
        for j in range(0,i+1):
          resden += dissim1(seq[i],seq[j])
    return resden


def fynplus1dissim(seq):
  return (dissimhelper1num(seq) / dissimhelper1den(seq))


def fynplus1(seq):
  return fynplus1dissim(seq)+fynplus1sim(seq)
```

```
seq = [1,2,3]

print(fynplus1(seq))


seq = [2,3,5,7,11,13,17,19]

print(fynplus1(seq))


92/21 #simdem

47/3 #simnum

(47/3)/(92/21) #simfynplus1   3.5760869565217397 verified

34/21 #dissimdem

25/3 #dissimnum

(25/3)/(34/21) #dissimfynplus1  5.147058823529411 verified


seq = [3,2,7]

print(fynplus1(seq))


seq = [19,17,13,11,7,5,3,2]
```

```
print(fynplus1(seq))


seq = [-5.1451,2,3,5,7,11,13,17,19]
print(fynplus1(seq))


# all subseries


seq = [2,3,5,7,11,13,17,19]
dupseq = seq.copy()
dupseq.append(0)
lst = []
for j in range(2,len(dupseq)+1):
  temp=[]
  for i in range(len(dupseq)-1,-1,-j):
    temp.append(dupseq[i])
  lst.append(temp)


# accessing only subseries with atleast 2 elements excluding 0
```

```
for i in range(len(lst)):
  if len(lst[i])<=2:
    inter = lst[:i]
    break

# removing 0

for i in inter:
  i.pop(0)

# reversing the subseries

for i in range(len(inter)):
  inter[i].reverse()

inter

# inserting the origial seq at the start
```

```
inter.insert(0,seq)
inter


sscosfa = []
for i in inter:
  sscosfa.append(fynplus1(i))


sscosfa


weights = []
for i in sscosfa:
  weights.append(i/sum(sscosfa))


weights


weightedaverage = 0
comb = list(zip(sscosfa,weights))
for i in comb:
```

```
    weightedaverage += (i[0]*i[1])
weightedaverage


# using a function


def highlyused(seq):
  dupseq = seq.copy()
  dupseq.append(0)
  lst = []
  for j in range(2,len(dupseq)+1):
    temp=[]
    for i in range(len(dupseq)-1,-1,-j):
      temp.append(dupseq[i])
    lst.append(temp)


# accessing only subseries with atleast 2 elements excluding 0
  for i in range(len(lst)):
    if len(lst[i])<=2:
```

```
      inter = lst[:i]
      break

# removing 0

  for i in inter:
    i.pop(0)

# reversing the subseries

  for i in range(len(inter)):
    inter[i].reverse()

# inserting the origial seq at the start
  inter.insert(0,seq)
  inter

  return inter
```

```
seq = [2,3,5,7,11,13,17,19]

fin = highlyused(seq)

sscosfa = []

for i in inter:

  sscosfa.append(fynplus1(i))


sscosfa
```